
UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

# ANÁLISIS DEL COMPORTAMIENTO DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES EN LA DETECCIÓN DE MALARIA

BORBOR MORENO DAYANARA MISHEL
INGENIERA DE SISTEMAS

MACHALA
2019


UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

# ANÁLISIS DEL COMPORTAMIENTO DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES EN LA DETECCIÓN DE MALARIA

BORBOR MORENO DAYANARA MISHEL
INGENIERA DE SISTEMAS

MACHALA
2019


# UTMACH

## FACULTAD DE INGENIERÍA CIVIL

## CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

# ANÁLISIS DEL COMPORTAMIENTO DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES EN LA DETECCIÓN DE MALARIA

BORBOR MORENO DAYANARA MISHEL
INGENIERA DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 21 DE AGOSTO DE 2019

MACHALA
21 de agosto de 2019

**Nota de aceptación:**

Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado ANÁLISIS DEL COMPORTAMIENTO DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES EN LA DETECCIÓN DE MALARIA, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente

RIVAS ASANZA WILMER BRAULIO
0702580192
TUTOR - ESPECIALISTA 1

NOVILLO VICUÑA JOHNNY PAUL
0702947409
ESPECIALISTA 2

HERNANDEZ ROJAS DIXYS LEONARDO
0923026298
ESPECIALISTA 3

Fecha de impresión: miércoles 21 de agosto de 2019 - 14:35

URKUND

# Urkund Analysis Result

| | |
|---|---|
| **Analysed Document:** | Informe para Urkund Dayanara.docx (D54801983) |
| **Submitted:** | 8/13/2019 4:00:00 PM |
| **Submitted By:** | wrivas@utmachala.edu.ec |
| **Significance:** | 0 % |

Sources included in the report:

Instances where selected sources appear:

0

## CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

La que suscribe, BORBOR MORENO DAYANARA MISHEL, en calidad de autora del siguiente trabajo escrito titulado ANÁLISIS DEL COMPORTAMIENTO DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES EN LA DETECCIÓN DE MALARIA, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

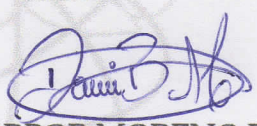
La autora declara que el contenido que se publicará es de carácter académico y se enmarca en las dispociones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

La autora como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 21 de agosto de 2019



BORBOR MORENO DAYANARA MISHEL
0750183113

## DEDICATORIA

En primer lugar, a Dios por brindarme la sabiduría, fortaleza y perseverancia para poder cumplir con mis objetivos en el transcurso de mis años de estudio y poder culminar ésta muy importante etapa en mi vida.

A mi familia, principalmente a mi madre que a pesar de las adversidades de la vida siempre me apoyó, por ser el claro ejemplo de una mujer luchadora que nunca se rinde, por confiar en mí y darme las fuerzas necesarias para seguir adelante. Mis hermanos quienes con cada sonrisa me motivaban para perseverar y alcanzar mis metas. Mi abuelita que siempre me brindó su ayuda, paciencia y nunca me negó su apoyo y cariño. A mi esposo que me comprendió en cada momento y dio su apoyo incondicional para finalizar esta etapa.

Dayanara Mishel Borbor Moreno

## AGRADECIMIENTO

Primero agradezco a Dios por bendecir cada paso dado, cada etapa de mi vida, por la fortaleza y perseverancia que cada día me otorga para salir adelante.

Mi familia, por ser mi motor, mi razón para continuar y no desmayar, por la comprensión y el apoyo incondicional que día a día me brindaron. A mis docentes por compartir su conocimiento en cada una de sus cátedras.

Dayanara Mishel Borbor Moreno

## RESUMEN

# ANÁLISIS DEL COMPORTAMIENTO DE ARQUITECTURAS DE REDES NEURONALES CONVOLUCIONALES EN LA DETECCIÓN DE MALARIA

Borbor Moreno Dayanara Mishel, 0750183113

La salud y la tecnología están muy relacionadas en la actualidad, ya que gracias a los avances tecnológicos se han logrado emplear diversas soluciones a los distintos problemas que se encuentran inmersos en la salud, uno de estos problemas es la detección de la malaria en un tiempo oportuno para su respectivo tratamiento. En el presente trabajo se realiza un aprendizaje profundo con los modelos Mobilenet e Inception V3, utilizando una base de datos de Kaggle llamada “Malaria Cell Images Dataset”, la cual está constituida por dos clases Parasitized que son las imágenes que presentan señales de existencia de la infección de Malaria y Uninfected que no presenta anomalías, cada clase contiene 13.780 imágenes dando un total de 27.560 imágenes, las herramientas tecnológicas utilizadas fueron Python como lenguaje de programación bajo su distribución Anaconda y entorno Jupyter y las librerías Keras y TensorFlow. Con ello se procedió a realizar un análisis del comportamiento de ambas arquitecturas, para la cual se consideró aproximadamente 8.000 imágenes, debido a que una gran cantidad de imágenes requiere de un equipo computacional muy potente y por la calidad de imágenes que contiene el dataset, ya que todas no poseen una buena resolución, por ende, la exactitud obtenida en ambos modelos fue de 0. 9894 y 0.9555, respectivamente.

**PALABRAS CLAVES:** Redes Neuronales Convolucionales, Mobilenet, Inception V3, Aprendizaje Profundo, Exactitud.

## ABSTRACT

### ANALYSIS OF THE BEHAVIOR OF ARCHITECTURES OF CONVOLUTIONAL NEURONAL NETWORKS IN THE DETECTION OF MALARIA

Borbor Moreno Dayanara Mishel, 0750183113

Health and technology are closely related today, because thanks to technological advances, various solutions have been achieved to the different problems that are immersed in health, one of these problems is the detection of malaria in a timely manner for their respective treatment. In this work, a deep learning is carried out with the Mobilenet and Inception V3 models, using a Kaggle database called "Malaria Cell Images Dataset", which is constituted by two Parasitized classes that are the images that show signs of existence of The infection of Malaria and Uninfected that does not present anomalies, each class contains 13,780 images that give a total of 27,560 images, the technological tools used were Python as a programming language under its Anaconda distribution and Jupyter environment and the Keras and TensorFlow libraries. With this, an analysis of the behavior of both architectures must be performed, for which approximately 8,000 images are considered, because a large number of images requires very powerful computational equipment and because of the quality of images contained in the data set, since it does not have a good resolution, therefore, the accuracy obtained in both models was 0.9894 and 0.9555, respectively.



**KEY WORDS:** Convolutional Neural Networks, Mobilenet, Inception V3, Deep Learning, Accuracy.

## CONTENIDO

## CONTENIDO DE ILUSTRACIONES



## CONTENIDO DE TABLAS

# 1. INTRODUCCIÓN

Actualmente la malaria es uno de los dilemas principales de salud a nivel mundial, ya que es el causante de al menos 80 millones de infectados por año alrededor del mundo. La mayoría de las infecciones suceden en África y siendo más vulnerables los infantes menores de 5 años. Cuyo principal reto es la identificación o diagnóstico y tratamiento pertinente. [1] [2]

Los avances tecnológicos han generado un gran impacto en algunas áreas, tales como la educación, agronomía, veterinaria, marketing, seguridad y por supuesto la salud no está exenta de esta gran revolución, brindado una gran ayuda a la resolución de problemas y obtener resultados precisos e inmediatos.

La inteligencia artificial está inmersa en esta revolución, brindando considerables ventajas, ya que al momento de detectar enfermedades en un tiempo oportuno se puede dar una solución a tiempo reduciendo gastos en tratamientos a diferencia de los tratamientos de enfermedades que son detectadas en una etapa avanzada.

En el presente trabajo se analizará el comportamiento de distintos modelos de Redes Neuronales Convolucionales, frente al aprendizaje de una base de datos obtenida en Kaggle llamada “Malaria Cell Images Dataset”, la cual posee alrededor de 27,560 imágenes. Las arquitecturas a evaluar son:

- Inception V3
- Mobilenet

Los diversos modelos de redes neuronales convolucionales generar diferentes resultados, tomando en cuenta la cantidad de imágenes y el problema al que es aplicada. La estructuración del presente informe se detalla a continuación:

Capítulo 1: Se presenta la introducción, su contexto de tal manera que permita manifestar y justificar la problemática y el objetivo a alcanzar con el desarrollo de este documento.

Capítulo 2: Se describen los puntos teóricos destinados como fundamento para el desarrollo de la resolución del problema planteado, detallando la solución obtenida.

Capítulo 3: Se detallan las conclusiones obtenidas basadas en los resultados generados, demostrando de esta manera el cumplimiento del propósito del documento.

### 1.1. Marco Contextual

El reconocimiento y clasificación de imágenes en la actualidad es realizado por redes neuronales convolucionales, debido a que son arquitecturas donde las neuronas son muy similares a la corteza visual de un cerebro humano, con respecto a la recepción de datos.

Existen diversas redes neuronales convolucionales, pero no todas garantizan resultados efectivos, ya sea por la cantidad de imágenes o por el campo en el que es empleada.

### 1.2. Problema

Realizar un análisis de los modelos de redes neuronales convolucionales en el campo de la salud como lo es la detección de la malaria, ya que presenta una alta exigencia de efectividad en sus resultados, para tomar las medidas necesarias en personas infectadas de malaria.

### 1.3. Objetivo General

Analizar el comportamiento de arquitecturas de redes neuronales convolucionales mediante la comparación de los resultados obtenidos con el aprendizaje de imágenes, para la identificación de imágenes infectadas y no infectadas de malaria.

# 2. DESARROLLO

## 2.1. Marco Teórico

### 2.1.1. Redes Neuronales Convolucionales

Las redes neuronales convolucionales es una arquitectura inspirada en el aprendizaje profundo, han manifestado ser eficaces en distintas áreas, como el reconocimiento o clasificación de personas según su género [3], animales microinvertebrados [4], medios de transporte, cáncer de mama, e identificación de tumores.

Sus capas totalmente interconectadas fusionan las salidas de las primeras capas en un alto nivel. Por ende, pueden aprender de manera automática peculiaridades de alto nivel, demostrando su eficiente capacidad de clasificación. [5]

### 2.1.2. Clasificación de imágenes

La clasificación de imágenes es una de los procesos primordiales, para que los elementos evaluados se puedan identificar mediante un video o imagen. Se relaciona directamente con el aprendizaje profundo, de tal manera que aumente la eficiencia de la clasificación de imágenes. [6]

### 2.1.3. Aprendizaje profundo

El aprendizaje profundo (Deep Learning) es un capo de inteligencia artificial, perteneciente al área de aprendizaje automático. Es caracterizada por la transformación de las características de los parámetros que extrae. [7]

### 2.1.4. Herramientas tecnológicas

#### 2.1.4.1. Python

Se caracteriza por su simplicidad para la programación orientada a objetos, funcional e imperativa, por tanto, es considerado multi-paradigmas. Siendo un lenguaje de alto nivel, facilita el desarrollo de complejas tareas, debido a que posee tuplas, listas y diccionarios. [8]

Los métodos de aprendizaje, han alcanzado un alto éxito con relación al reconocimiento de voz, detección de elementos, contribuyendo con alto rendimiento a fines industriales y académicos. [9]

#### 2.1.4.2. TensorFlow

TensorFlow es una arquitectura de aprendizaje muy conocida y utilizada de inteligencia artificial. Es de Google de segunda generación, basada en DistBelief. Este sistema es una biblioteca de software de código libre, que para sus respectivos cálculos usa esquemas de flujo de datos. [10]

### 2.1.5. Modelos de redes neuronales

#### 2.1.5.1. Inception V3

Publicado por GoogleNet, se le realizaron cambio que permitieron la optimización de los resultados del modelo al aumentar el flujo de la información, se equiparo la profundidad y el número de filtros en cada una de sus etapas. [11]

#### 2.1.5.2. Mobilenet

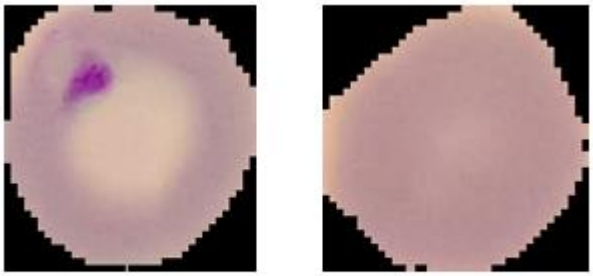
Mobilenet es modelo muy ligero que permite a las redes neuronales convolucionales de una manera eficaz. Su arquitectura se construye usando una convolución separada en profundidad. [12]

## 2.2. Marco Metodológico

Para el presente documento se utilizó un dataset de Kaggle llamada “Malaria Cell Images Dataset”, la cual posee alrededor de 27,560 imágenes, de las cuales se utilizó 8000 imágenes cada una con dimensiones diferentes de aproximado (130x133) Las clases que posee está base de datos son 2:

1. Parasitized: Presenta el virus de la malaria
2. Uninfected: No presenta anomalías

*Figura 1 Apariencia de las clases que posee el Dataset*




*Fuente: [13]*

Para su aprendizaje profundo se utilizó dos modelos diferentes y muy conocidos Incepption-V3 y Mobilenet, siendo las más utilizadas al momento de clasificar imágenes. Los parámetros de cada modelo se detallan en la Tabla 1.

*Tabla 1 Cantidad de parámetros de cada modelo*

| **Modelos** | **Parámetros de los modelos de cada Red Neuronal Convolucional** | | |
|---|---|---|---|
| | **Parámetros entrenables** | **Parámetros no entrenables** | **Parámetros totales** |
| Inception V3 | 25,441,954 | 34,432 | 25,476,386 |
| Mobilenet | 5,832,002 | 21,888 | 5,853,890 |

*Fuente: Elaboración propia*

Ambos modelos se utilizaron para el aprendizaje y clasificación, la base de datos posee dos clases, de tal manera que la primera clase pertenece a las imágenes con señales de infección de malaria y la segunda clase no presenta anomalías.

La tabla 2 indica el número de imágenes que posee cada clase y la cantidad de imágenes utilizadas para el desarrollo este trabajo.

*Tabla 2 Detalle de la cantidad de imágenes utilizadas*

| **Clases** | **Número total de imágenes** | **Número de imágenes utilizadas** |
|---|---|---|
| Clase 1: Parasitized | 13,780 | 4,000 |
| Clase 2: Uninfected | 13,780 | 4,000 |

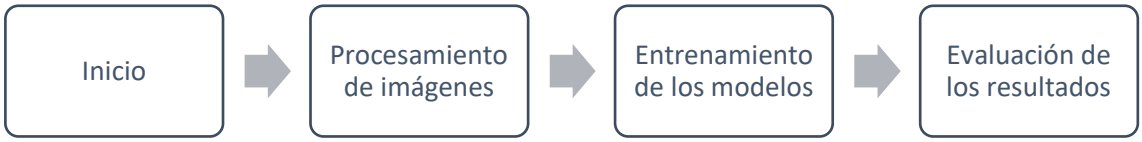
*Fuente: Elaboración propia*

El dataset utilizado consta de imágenes de diferente dimensión y apariencia, además de esto posee imágenes con iluminación en las de la clase no infectadas

y en las infectadas hay imágenes que se logran observar su presencia de la infección, pero esta es muy leve.

Cada red neuronal fue entrenada por separado, cuyo entrenamiento presentó un alto costo computacional, debido al tiempo que tarda cada red en aprender y por la cantidad de imágenes que aprende.

*Figura 2 Ciclo del aprendizaje de los modelos*




*Fuente: Elaboración propia*

El desarrollo de una arquitectura basada en redes neuronales convolucionales existen diferentes maneras de llevarlas a cabo, debido a que en la actualidad hay frameworks o API's expertos en redes neuronales. El presente trabajo se llevó a cabo con las librerías TensorFlow y Keras, como lenguaje de programación Python, bajo la distribución de Anaconda y su entorno Jupyter porque permite un trabajo interactivo.

El Hardware utilizado posee una tarjeta de video NVIDIA GTX 960M de 2GB y procesador Intel Core i7 de 16GB de RAM de sexta generación.

### 2.3. Resultados

Se logró como resultados lo que se detalla en la tabla 3. Se puede observar el Accuracy(exactitud), loss (pérdida) y el tiempo que tomó el entrenamiento. La prueba fue realizada con imágenes de ambas clases infectadas y no infectadas.

*Tabla 3 Resultados obtenidos del aprendizaje de los modelos*

| Modelos | Accuracy (Exactitud) | Loss (Pérdida) | Tiempo |
|---|---|---|---|
| Inception V3 | 0.9555 | 0.1238 | 71h56m54s |
| Mobilenet | 0.9894 | 0.0331 | 69h24m34s |

*Fuente: Elaboración propia*

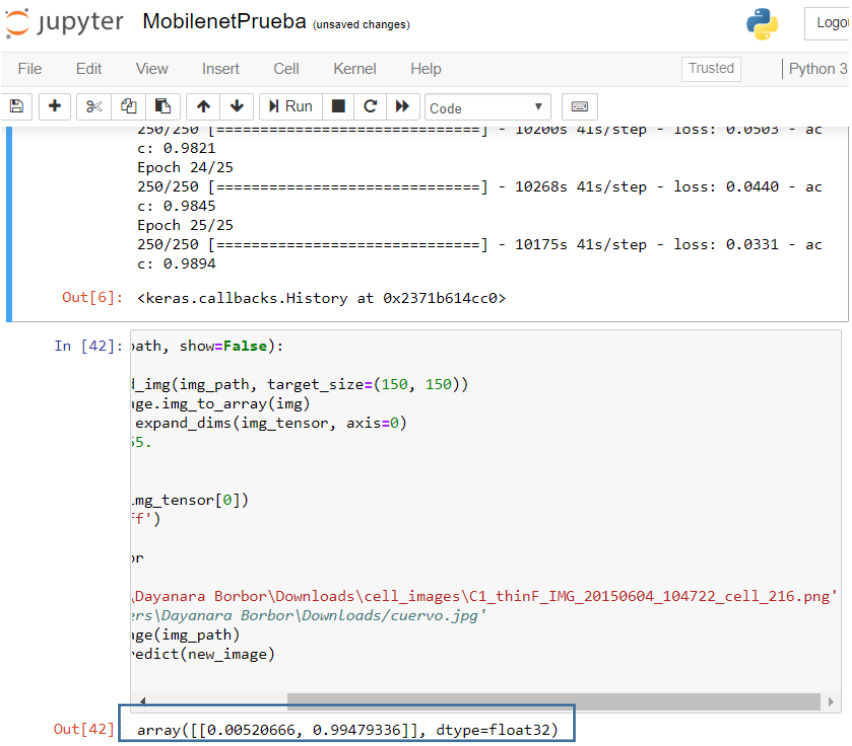
Al momento de evaluar la imagen que se muestra a continuación, la cual pertenece a la clase Uninfected (No infectado):

*Figura 3 Imagen de la clase Uninfected*


*Fuente: [13]*

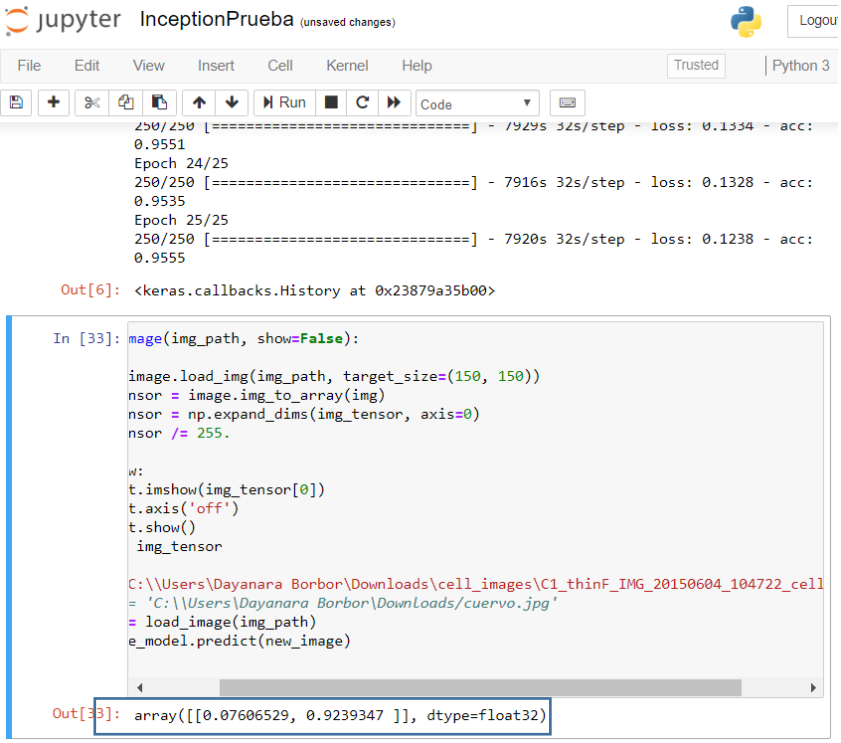
Mobilenet indica que es una imagen que pertenece a la clase Uninfected con un 0.99479336, de igual manera Inception V3 indicó que pertenece a la clase Uninfected con un 0.9239347, como se puede observar en la imagen a continuación:

*Figura 4 Prueba realizada con el modelo Mobilenet*




*Fuente: Elaboración propia*

*Figura 5 Prueba realizada con el modelo Inception V3*




*Fuente: Elaboración propia*

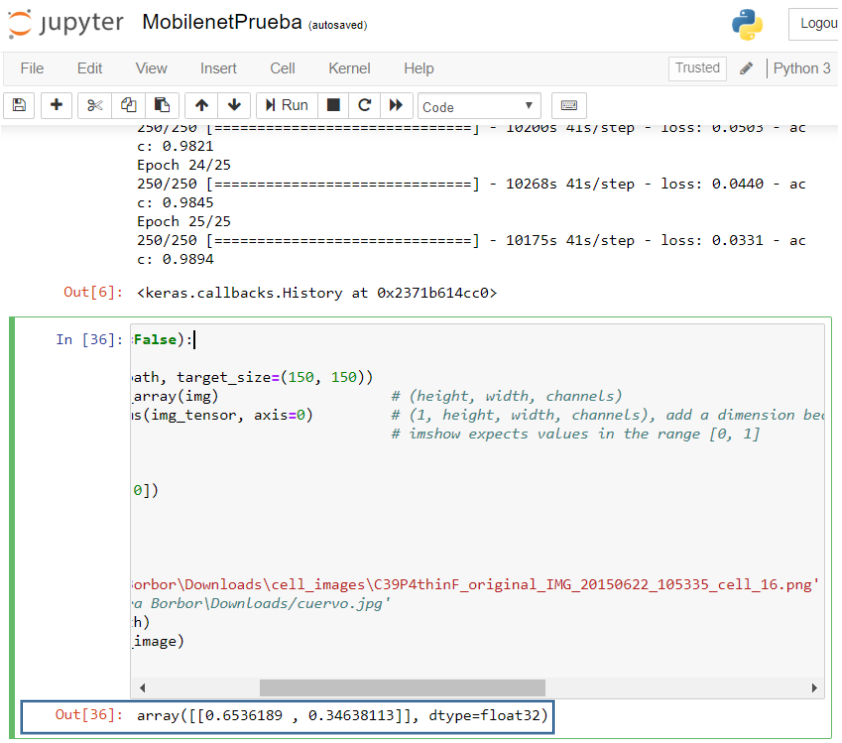
Ahora se evalúa la siguiente imagen, perteneciente a la clase Parasitized:

*Figura 6 Imagen de clase Parasitized*


*Fuente: [13]*

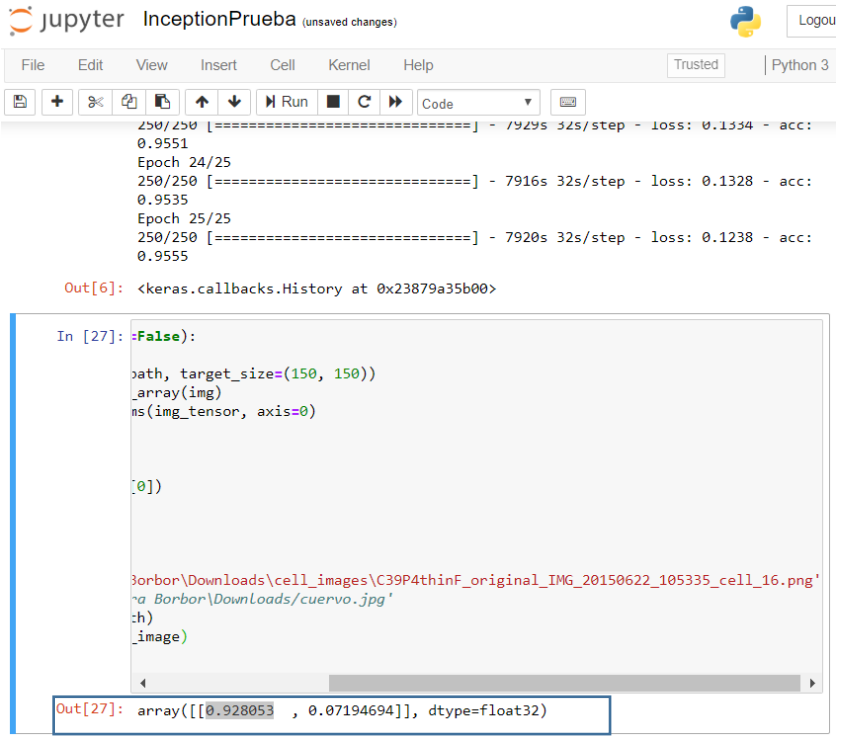
En las imágenes a continuación se logra observar que ambos modelos indican que es una imagen de clase Parasitized. Pero Inception V3 da un resultado de 0.928053 siendo el mejor resultado.

*Figura 7 Prueba con el modelo Mobilenet*




*Fuente: Elaboración propia*

*Figura 8 Prueba con el modelo Inception V3*




*Fuente: Elaboración propia*

## 3. CONCLUSIONES

Se logró observar que al momento de utilizar diferentes arquitecturas de redes neuronales convolucionales generan resultados distintos a pesar de que los parámetros a entrenar sean los mismo, como la cantidad de imágenes, el optimizador ADAM, por ello se debe realizar la prueba con diferentes modelos, para así poder determinar cuál es el más óptimo, según el problema al que se emplea.

En los modelos evaluados se logró constatar que su estructura se enfatiza unas más que otras y esto no se da sólo por su exactitud o pérdida, también se da por la cantidad de parámetros que se van a entrenar, ya que esto requiere que la exigencia del recurso Hardware sea más alta.

El análisis del comportamiento entre Mobilenet e Inception V3, se pudo comprobar que en este caso de la detección de Malaria el que generó un mejor rendimiento al momento de realizar las pruebas fue Inception V3, ya que dio resultados óptimos, en cambio Mobilenet se logró observar que tiene falencias cuando se trata de detectar imágenes que pertenecen al grupo Parasitized.

# 4. BIBLIOGRAFÍA


[1] M. A. Aristizábal Giraldo, L. M. Martínez Sánchez y D. A. Quintero Moreno, «Malaria, enfermedad tropical de múltiples métodos diagnósticos,» *Archivos de Medicina (Manizales),* vol. 17, nº 2, pp. 402-414, 2017.

[2] M. V. Valero Bernal, M. Tanner, S. Muñoz Navarro y J. F. Valero Bernal, «Proportion of fever attributable to malaria in Colombia:Potential indicators for monitoring progress towards malaria elimination,» *Revista de Salud Pública,* vol. 19, nº 1, 2017.

[3] A. S. Suárez L, A. F. Jiménez L, M. Castro Franco y A. Cruz Roa, «Clasificación y mapeo automático de coberturas del suelo en imágenes satelitales utilizando Redes Neuronales Convolucionales,» *Orinoquia,* vol. 21, nº 1, pp. 64-75, 2017.

[4] C. Quintero, F. Merchán, A. Cornejo y J. Sánchez Galán, «Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitoreo Participativo,» *6th Engineering, Science and Technology Conference,* vol. 2018, pp. 585-596, 2017.

[5] Y. Zhou, X. Liu, J. Zhao, D. Ma, R. Yao, B. Liu y Y. Zheng, «Remote sensing scene classification based on rotation-invariant feature learning and joint decision making,» *EURASIP Journal on Image and Video Processing,* vol. 2019, nº 3, p. 3, 2019.

[6] D. Plotnikov, E. Sopov y I. Panfilov, «An approach for automating the design of convolutional neural networks,» *IOP Conf. Series: Materials Science and Engineering,* vol. 450, nº 5, 2018.

[7] J. F. Puerta Barrera, D. Amaya Hurtado y R. Jiménez Moreno, «Prediction system of erythemas for phototypes I and II, using deep-learning,» *Pharmacology and Toxicology,* vol. 22, nº 3, 2015.

[8] I. Challenger Pérez, Y. Díaz Ricardo y R. Becerra García, «El lenguaje de programación Python,» *Ciencias Holguín,* vol. XX, nº 2, pp. 1-13, 2014.

[9] F. Hu, G. Xia, J. Hu y L. Zhang, «Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery,» *Remote Sens,* vol. 7, pp. 14680-14707, 2015.

[10] L. Shi, Z. Li y D. Song, «A Flower Auto-Recognition System Based on Deep Learning,» *IOP Conf. Series: Earth and Environmental Science,* vol. 234, 2019.

[11] L. ZhanLi y Y. JiaWei, «Abnormal Behavior Recognition Based on Transfer Learning,» *IOP Conf. Series: Journal of Physics: Conf. Series ,* vol. 1213, 2019.

[12] Q. Niu, Y. Teng y L. Chen, «Design of gesture recognition system based on Deep Learning,» *IOP Conf. Series: Journal of Physics: Conf. Series ,* vol. 1168, 2019.

[13] Arunava, «Kaggle,» 05 12 2018. [En línea]. Available: https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria. [Último acceso: 02 08 2019].

## ANEXOS

**ANEXO A:** Se logra observar que como primera instancia se deben cargar las librerías y paquetes necesarios

*Figura 9 Carga de librerías y paquetes del modelo Inception V3*

```
import keras
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense,GlobalAveragePooling2D
from keras.preprocessing import image
from keras.models import Model
from IPython.display import Image
from keras.applications import imagenet_utils
import numpy as np
from keras.metrics import categorical_crossentropy
from keras.optimizers import Adam
from keras.metrics import categorical_crossentropy
from keras.layers.core import Dense, Activation,Flatten
from keras import backend as K
from IPython.display import Image
from keras.optimizers import Adam
from keras.applications.inception_v3 import InceptionV3
from keras.applications.inception_v3 import preprocess_input, decode_predictions
```

```
Using TensorFlow backend.
```

*Fuente: Elaboración propia*

*Figura 10 Carga de librerías y paquetes del modelo Mobilenet*

```
import keras
from keras import backend as K
from keras.layers.core import Dense, Activation
from keras.optimizers import Adam
from keras.metrics import categorical_crossentropy
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from keras.models import Model
from keras.applications import imagenet_utils
from keras.layers import Dense,GlobalAveragePooling2D,Flatten
from keras.applications import MobileNet
from keras.applications.mobilenet import preprocess_input
import numpy as np
from IPython.display import Image
from keras.optimizers import Adam
```

```
Using TensorFlow backend.
```

*Fuente: Elaboración propia*

**ANEXO B:** Se procede a agregar capas y entrenar las capas superiores, detallando el número de clases con las que se va a trabajar.

*Figura 11 Agregación de capas al modelo Inception V3*

```
base_model=InceptionV3(weights=None,include_top=False, classes=2) #imports the m
x=base_model.output
x=GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x) #we add dense layers so that the model can le
x=Dense(1024,activation='relu')(x) #dense layer 2
x=Dense(512,activation='relu')(x) #dense layer 3
preds=Dense(2,activation='softmax')(x)
```

*Fuente: Elaboración propia*

*Figura 12 Agregación de capas al modelo Mobilenet*

```
base_model=MobileNet(weights=None,include_top=False, classes=2) #imports the mob
x=base_model.output
x=GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x) #we add dense layers so that the model can le
x=Dense(1024,activation='relu')(x) #dense layer 2
x=Dense(512,activation='relu')(x) #dense layer 3
preds=Dense(2,activation='softmax')(x)
```

*Fuente: Elaboración propia*

**ANEXO C:** Se realizar el resumen del modelo para observar el número de parámetros a entrenar.

*Figura 13 Resumen del modelo Inception V3*

```
base_model=Model(inputs=base_model.input,outputs=preds)
base_model.summary()
```

```
_________________________
dense_2 (Dense)              (None, 1024)              1049600     dense_1
[0][0]
__________________________________________________________________________________
_________________________
dense_3 (Dense)              (None, 512)               524800      dense_2
[0][0]
__________________________________________________________________________________
_________________________
dense_4 (Dense)              (None, 2)                 1026        dense_3
[0][0]
==================================================================================
========================
Total params: 25,476,386
Trainable params: 25,441,954
Non-trainable params: 34,432
__________________________________________________________________________________
_________________________
```

*Fuente: Elaboración propia*

*Figura 14 Resumen del modelo Mobilenet*

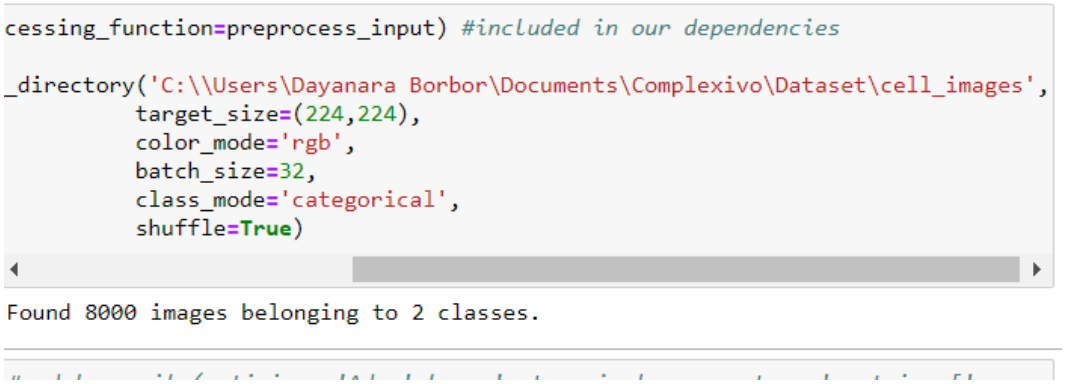
```
base_model=Model(inputs=base_model.input,outputs=preds)
base_model.summary()
```

```
_________________________________________________________________
conv_pw_13_bn (BatchNormaliz (None, None, None, 1024)  4096
_________________________________________________________________
conv_pw_13_relu (ReLU)       (None, None, None, 1024)  0
_________________________________________________________________
global_average_pooling2d_2 ( (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600
_________________________________________________________________
dense_3 (Dense)              (None, 512)               524800
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 1026
=================================================================
Total params: 5,853,890
Trainable params: 5,832,002
Non-trainable params: 21,888
_________________________________________________________________
```

*Fuente: Elaboración propia*

**ANEXO D:** Carga de las imágenes para el entrenamiento, especificando la ruta.

*Figura 15 Carga de datos para el entrenamiento del Modelo Inception V3*

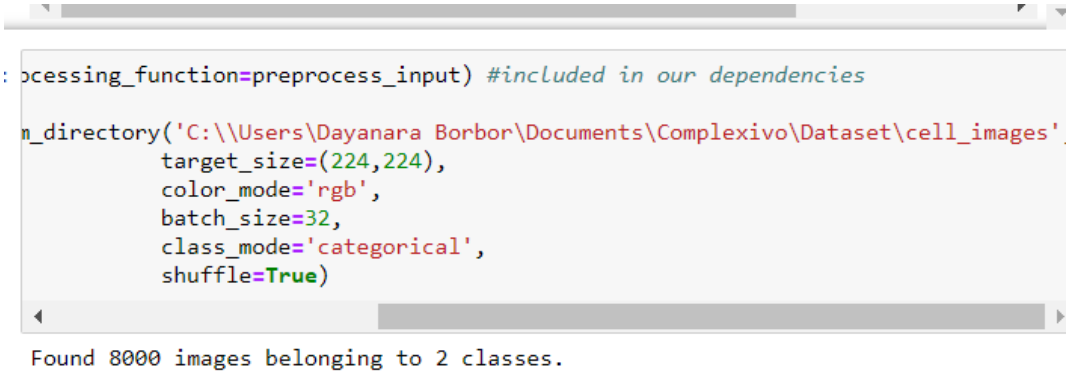


```
cessing_function=preprocess_input) #included in our dependencies

_directory('C:\\Users\Dayanara Borbor\Documents\Complexivo\Dataset\cell_images',
          target_size=(224,224),
          color_mode='rgb',
          batch_size=32,
          class_mode='categorical',
          shuffle=True)
```

```
Found 8000 images belonging to 2 classes.
```

*Fuente: Elaboración propia*

*Figura 16 Carga de datos para el entrenamiento del Modelo Mobilenet*



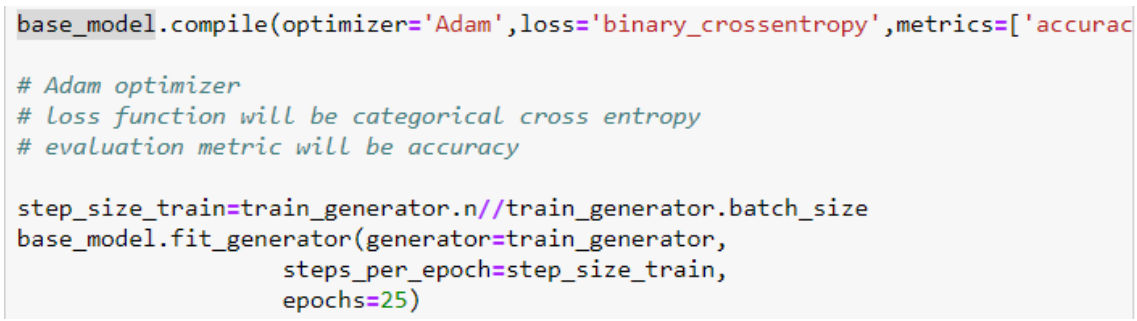
```
ocessing_function=preprocess_input) #included in our dependencies

n_directory('C:\\Users\Dayanara Borbor\Documents\Complexivo\Dataset\cell_images'
          target_size=(224,224),
          color_mode='rgb',
          batch_size=32,
          class_mode='categorical',
          shuffle=True)
```

```
Found 8000 images belonging to 2 classes.
```

*Fuente: Elaboración propia*

**ANEXO E:** Entrenamiento de los modelos utilizando el optimizador ADAM.

*Figura 17 Entrenamiento del modelo Inception V3*



```
base_model.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['accurac

# Adam optimizer
# loss function will be categorical cross entropy
# evaluation metric will be accuracy

step_size_train=train_generator.n//train_generator.batch_size
base_model.fit_generator(generator=train_generator,
                   steps_per_epoch=step_size_train,
                   epochs=25)
```

*Fuente: Elaboración propia*

*Figura 18 Entrenamiento del modelo Mobilenet*

```
base_model.compile(optimizer='Adam',loss='binary_crossentropy',metrics=['accurac

# Adam optimizer
# loss function will be categorical cross entropy
# evaluation metric will be accuracy

step_size_train=train_generator.n//train_generator.batch_size
base_model.fit_generator(generator=train_generator,
                   steps_per_epoch=step_size_train,
                   epochs=25)
```

*Fuente: Elaboración propia*

## **ANEXO F:** Resultados obtenidos del entrenamiento

*Figura 19 Resultados obtenido del entrenamiento del modelo Inception V3*

```
Epoch 23/25
250/250 [==============================] - 7929s 32s/step - loss: 0.1334 - acc:
0.9551
Epoch 24/25
250/250 [==============================] - 7916s 32s/step - loss: 0.1328 - acc:
0.9535
Epoch 25/25
250/250 [==============================] - 7920s 32s/step - loss: 0.1238 - acc:
0.9555

<keras.callbacks.History at 0x23879a35b00>
```

*Fuente: Elaboración propia*

*Figura 20Resultados obtenidos del entrenamiento del modelo Mobilenet*

```
Epoch 23/25
250/250 [==============================] - 10200s 41s/step - loss: 0.0503 - ac
c: 0.9821
Epoch 24/25
250/250 [==============================] - 10268s 41s/step - loss: 0.0440 - ac
c: 0.9845
Epoch 25/25
250/250 [==============================] - 10175s 41s/step - loss: 0.0331 - ac
c: 0.9894

<keras.callbacks.History at 0x2371b614cc0>
```

*Fuente: Elaboración propia*